

MODERNIZING LIME CRM WEB CODE

OVERVIEW

1. Background
2. Current state of code
3. Future state of code
4. Plan
5. Execution
6. Risks
7. After base renovations
8. Chosen solutions

1. BACKGROUND

- Ongoing project for years
- Lots of features implemented
- Prio new features and bug fix
- Quite a bit of technical debt
- Wish to achieve a modern JS-code base
- Rewrite from scratch? Migrate?

2. CURRENT STATE OF CODE

- Automated tests?
- What build system is used?
- Dependency management
- Modular code?
- Hand written solutions?

3. FUTURE STATE OF CODE

- Mantra: simplify!
- Browser support nowadays?
- What can be removed?
- Simplest possible build system
- Standardized code formatting

4. PLAN

- Clear goals / milestones
- Separated iterations
- Where to begin
 - Lots of value for little effort

5. EXECUTION

- Begin with build system, easy to verify
- Modularise without reorganizing code
- Keep behaviors and names of build scripts
- Test constantly - both auto and manually
- Always do only one thing at a time!

6. RISKS

- Doing more than one thing at a time
 - Rollback needs to be trivial
- Replacing a complex thing with another complex thing
- Rewriting something "just because"

7. NEXT STEPS?

- Modules and modern JS available
- AngularJS -> Angular (5)
- State management with Redux
- Web Components
- Componentized UI-components

CHOSEN SOLUTIONS



JAVASCRIPT-MODULES



- Newly standardized
 - Predecessors AMD, CommonJS
- Code navigation in editor
- Autocomplete, intellisense

COMPILE TO JAVASCRIPT



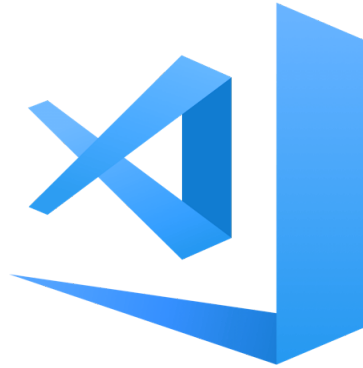
- TypeScript: Superset of JavaScript
 - Transpiles modern JS to work in browsers
 - Refactoring no-brainer

DATA STORE



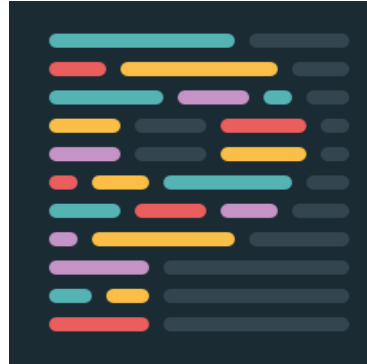
- All data in one place
- Easy to organize and test
- Time traveling - "undo"

EDITOR WITH CODE ANALYSIS



- Visual Studio Code
- Atom
- VIM
- Sublime
- Autocompletion etc.

CODE FORMATTING

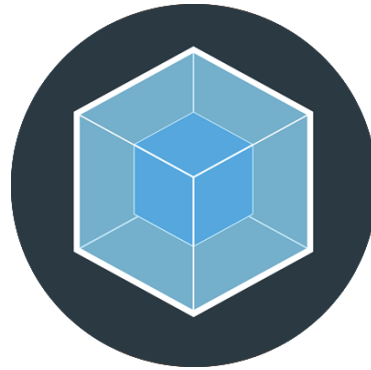


- Prettier.js
- Opinionated
- Can set a few options
- Automatic formatting
 - Scary until you've tried



- Scaffolding base and components
- All build steps necessary
- Environments

WEBPACK (VIA ANGULAR CLI)



- Automated bundling and building JS
- Also CSS-moduler etc.
- Live reload
- Cache busting, minifying code

FRONT END FRAMEWORK



- Complete FE Framework
- Components for UI element
- FormBuilder, Router, HTTP Client
- Observables, One way binding